

AD-A042 648

ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 9/4
AN EFFICIENT ALGORITHM FOR THE SPECTRA OF BLOCK CODED PAM SIGNA--ETC(U)
JUN 77 V DI EULIIS

UNCLASSIFIED

R-771

DAAB07-72-C-0259

NL

1 OF 1
AD
A042648

END
DATE
FILMED


8-77

DDC

ADA 042648

REPORT R-771 JUNE, 1977

UILU-ENG 77-2218

 **COORDINATED SCIENCE LABORATORY**

12

**AN EFFICIENT ALGORITHM
FOR THE SPECTRA
OF BLOCK CODED
PAM SIGNALS**

VAL DI EULIIS

DDC
RECEIVED
AUG 9 1977
D

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

AD No. _____
DDC FILE COPY.

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN EFFICIENT ALGORITHM FOR THE SPECTRA OF BLOCK CODED PAM SIGNALS		5. TYPE OF REPORT & PERIOD COVERED 9 Technical Report,
6. AUTHOR(s) Val Di Euliis <i>see next page</i>		6. PERFORMING ORG. REPORT NUMBER 14 R-771, UIBU-ENG-77
7. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, Illinois 61801		8. CONTRACT OR GRANT NUMBER(s) 15 DAAB-07-72-C-0259, NSF-ENG-75-22621
9. CONTROLLING OFFICE NAME AND ADDRESS Joint Services Electronics Program		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE Jun 1977
		13. NUMBER OF PAGES 22 <i>12 276</i>
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Spectral Analysis Line Coding		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The power spectral density of multi-level coded sequences has been an important factor in the design of PCM transmission schemes. The spectral density for symbol-by-symbol encoding schemes was first derived by Bennett. In the past few years, several algorithms for block encoded schemes have been proposed. The most recent algorithm is based on a generalization of Bennett's formula and a compact notation for finite state machine encoders. This paper derives closed form expressions for the covariance sequences of multi-level block coded sequences and relates these covariance coefficients to Bennett's formula		

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

097700

Donec

OVER

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. ABSTRACT (continued)

such that the ensuing algorithm is more efficient in time and memory.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ACCESSION for	
NTIS	White Section <input checked="" type="checkbox"/>
DDC	Buff Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	

UILU-ENG 77-2218

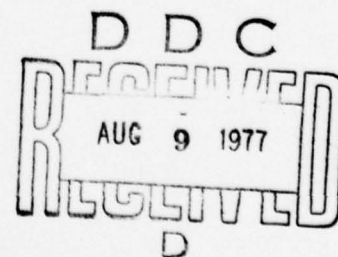
AN EFFICIENT ALGORITHM FOR THE SPECTRA OF BLOCK CODED PAM SIGNALS

by

Val DiEuliis

This work was supported by the National Science Foundation under Grant NSF ENG 75-22621; additional support was provided by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract DAAB-07-72-C-0259.

Reproduction in whole or in part is permitted for any purpose of the United States Government.



<p>DISTRIBUTION STATEMENT A</p> <p>Approved for public release; Distribution Unlimited</p>

Abstract

The power spectral density of multi-level coded sequences has been an important factor in the design of PCM transmission schemes. The spectral density for symbol-by-symbol encoding schemes was first derived by Bennett. In the past few years, several algorithms for block encoded schemes have been proposed. The most recent algorithm is based on a generalization of Bennett's formula and a compact notation for finite state machine encoders. This paper derives closed form expressions for the covariance sequences of multi-level block coded sequences and relates these covariance coefficients to Bennett's formula such that the ensuing algorithm is more efficient in time and memory.

I. Introduction

The power spectral density of multi-level coded sequences has been an important factor in the design of PCM transmission schemes. In [1], Bennett derived the formula for the spectrum of a symbol by symbol encoding scheme. An algebraic method for computing the spectrum was given by Tsujii and Kasai in [2]. In [3], Yasuda describes a more direct method of computation. Bosik [4] derived expressions for the autocorrelation sequence of a block coded signal, but did not show a closed form expression for the power spectral density. In [5], Cariolaro and Tronca develop an efficient closed-form algorithm for the computation of the spectral density of multi-level block codes with encoders described as finite state machines.

This paper derives closed form expressions for the covariance sequences of multi-level block codes and relates the covariance coefficients to Bennett's formula. In many ways the development follows [5], however a more efficient notation which may be considered as a generalization of [6], yields an algorithm which is computationally more efficient in time and memory.

II. Definitions and Notations

The class of encoders considered in this report may be described by means of a finite state machine. Let the quintuple $\mathcal{A} = \{\mathcal{I}, \mathcal{O}, \mathcal{S}, f, g\}$ denote the encoder with $\mathcal{I} = \{0, 1\}^K$, inputs consisting of sequences of K binary symbols (input blocks), $\mathcal{O} \subset L^N$, outputs consisting of sequences of N L -ary symbols* (output blocks or codewords), ($N \leq K$), and $\mathcal{S} = \{1, 2, \dots, I\}$, a finite set of states conveniently denoted as integers. The output function, $f: \mathcal{I} \times \mathcal{S} \rightarrow \mathcal{O}$ can be described as a set of functions $F = \{f_1, f_2, \dots, f_I\}$ where $f_i: \mathcal{I} \rightarrow \mathcal{O}$. The mapping f_i shall be referred to as the encoding mode corresponding to state i . Furthermore, we shall call a codeword $\underline{a} = (a^{(1)}, a^{(2)}, \dots, a^{(N)})$, with each $a^{(i)} \in L$, an element of state i if $f_i(b) = \underline{a}$ for some $b \in \mathcal{I}$.

The input block sequence is assumed to be stationary (i.e., each distinct block of K bits properly framed) with each block statistically independent. For each b_u ; $u = 1, \dots, 2^K \triangleq Q$, its associated probability will be denoted q_u . Of course we have the condition $\sum_{u=1}^Q q_u = 1$. This binary random process is transformed into an L -ary random process by the encoder. The encoded process may be represented as in Fig. 1 below.

$$\dots | \overset{s_0}{a_0^{(1)} a_0^{(2)} \dots a_0^{(N)}} | \overset{s_1}{a_1^{(1)} a_1^{(2)} \dots a_1^{(N)}} | \dots | \overset{s_k}{a_k^{(1)} a_k^{(2)} \dots a_k^{(N)}} | \dots$$

Figure 1. Channel sequence

*The elements of L shall be considered integers with $L = \{\ell \mid |\ell| \leq |L| - 2\}$ for L odd, and $L = \{2\ell \mid |\ell| \leq \frac{1}{2}|L| \text{ and } \ell \neq 0\}$ for $|L|$ even.

This representation shows the symbol sequence properly framed with respect to the words of length N . The symbol $a_k^{(\ell)}$ represents the ℓ^{th} digit of the k^{th} codeword in the channel sequence. We will refer to the time of transition from word \underline{a}_i to \underline{a}_{i+1} (i.e., the symbol transition $a_i^{(N)}$ to $a_{i+1}^{(1)}$) as wordtime i . Note that the sequence representation explicitly shows the state sequence of the encoder. We shall also adopt the convention that the state transition from s_i to s_{i+1} occurs at wordtime i .

At wordtime n , a new binary block is input to the encoder and the state transition function changes the internal state of the encoder according to $s_{n+1} = g(s_n, b_n)^*$. Under the assumption that the binary input process is stationary with statistically independent elements, it is easily shown that the state sequence is a homogeneous Markov chain. Let B_{ij} be the set of input words which cause the encoder to change from state i to state j [i.e., $B_{ij} = \{b \in \mathcal{B} \mid j = g(i, b)\}$] and denote the one-step transition matrix of the state process as $\Pi = \|\pi_{ij}\|$; then

$$\pi_{ij} = \sum_{u \in B_{ij}} q_u.$$

A restriction we place on Π (hence, the state transition function, g) is that the state sequence be a fully regular homogeneous Markov chain. For this class of state transition matrices, there exists $\Pi_\infty \triangleq \lim_{n \rightarrow \infty} \Pi^n$ where

$\Pi^n = \|\pi_{ij}^{(n)}\|$ denotes the matrix of n -step transition probabilities.

*The symbols s_n , and b_n shall always be interpreted as the state of the encoder at wordtime n , and the input block at time n respectively.

Furthermore, Π_∞ consists of identical rows $p = [p_1, \dots, p_I]$, such that $p\Pi = p$.

The vector p is called the stationary probability distribution.

The channel symbol sequence is cyclo-stationary with period N .

Hence the mean of the process is periodic with period N . Let $\alpha_{iu}^{(\ell)} \in L$

denote the ℓ^{th} digit of the codeword in state i corresponding to input word

b_u . Assuming the encoder to be in state i , the conditional mean,

$E[a_n^{(\ell)} | s_n = i]$, is given by:

$$E[a_n^{(\ell)} | s_n = i] = \sum_{u=1}^Q \alpha_{iu}^{(\ell)} q_u = \sum_{j=1}^I \left[\sum_{u \in B_{ij}} \alpha_{iu}^{(\ell)} q_u \right].$$

For reasons which will become clear later, let us define

$$m_{ij}^{(\ell)} \triangleq \sum_{u \in B_{ij}} \alpha_{iu}^{(\ell)} q_u$$

as the ℓ^{th} digit transition weight from state i to state j , and let the

matrix $M_\ell = \|m_{ij}^{(\ell)}\|; i, j = 1, \dots, I$. Thus,

$$E[a_n^{(\ell)} | s_n = i] = \sum_{j=1}^I m_{ij}^{(\ell)}.$$

After averaging over the ensemble of states we arrive at

$$\mu^{(\ell)} \triangleq E[a_n^{(\ell)}] = \sum_{i=1}^I p_i \sum_{j=1}^I m_{ij}^{(\ell)} = p M_\ell \underline{u}$$

where \underline{u} is the $(I \times 1)$ column vector of 1's.

Since we are ultimately interested in obtaining an efficient algorithm to compute the power spectral density of the random sequence, we

shall need an efficient representation for the covariance sequence

$v_k: -\infty \leq k \leq \infty$ where $v_k = E[(a^{(j)} - \mu^{(j)})(a^{(j+k)} - \mu^{(j+k)})]$ is the

covariance for elements in the encoded sequence which are separated by

(k-1) symbols. The dependence of v on the time reference j is not included as the covariance v_k is calculated as the arithmetic mean of the N symbol covariances for $j=1, \dots, N$. This tacitly assumes that the symbol sequence is wide-sense stationary since an observer would not know the proper framing of the symbol sequence.

The covariance coefficients shall be represented by the contribution to v_k from within a single codeword and the contribution from different codewords in the channel sequence. For example

$$\begin{aligned}
 v_0 &= \frac{1}{N} \sum_{s=1}^N E[(a^{(s)} - \mu^{(s)})^2] \\
 &= \frac{1}{N} \sum_{i=1}^I \sum_{u=1}^Q \sum_{s=1}^N \{(\alpha_{iu}^{(s)} - \mu^{(s)})^2\} p_i q_u \\
 &= \frac{1}{N} \sum_{i=1}^I \sum_{u=1}^Q \sum_{s=1}^N \{\alpha_{iu}^{(s)} \alpha_{iu}^{(s)} - 2 \mu^{(s)} \alpha_{iu}^{(s)} + (\mu^{(s)})^2\} p_i q_u \\
 &= \frac{1}{N} \left[\left(\sum_{i=1}^I \sum_{u=1}^Q \sum_{s=1}^N (\alpha_{iu}^{(s)})^2 p_i q_u \right) - \sum_{t=1}^N (\mu^{(t)})^2 \right]. \quad (2.1)
 \end{aligned}$$

This expression was developed as follows: assuming the encoder to be in state i , each digit of each codeword (minus the digit mean) is correlated with itself, then these quantities are averaged over the ensemble of states.

The same representation can be used to calculate v_1 , where each symbol is correlated with its adjacent symbol within the codeword. However, when correlating the last digit with its adjacent symbol, we find ourselves concerned with the next codeword in the sequence. This leads to our consideration of the coefficients being comprised of two terms; a term representing the contribution to the coefficient from within a single

codeword, and a term representing the contribution of the correlation between symbols in different codewords. Furthermore, it should be easily seen that a contribution from within a codeword is non-zero only for coefficients v_0, \dots, v_{N-1} . Therefore, we shall represent v_k , $k = 1, \dots, N-1$ as

$$v_k = \frac{1}{N} (v'_k + v''_k)$$

with v'_k being the contribution from within a single codeword and v''_k being the contribution from adjacent codewords in the sequence.

The component v'_k is a direct generalization of (2.1), and is expressed as

$$v'_k = \sum_{i=1}^I \sum_{u=1}^Q \sum_{s=1}^{N-k} \alpha_{iu}^{(s)} \alpha_{iu}^{(s+k)} q_u p_i - \left(\sum_{t=1}^{N-k} \mu(t) \mu(t+k) \right). \quad (2.2)$$

To simplify notation we shall define the N column vectors, \underline{W}_k , ($k = 0, \dots, N-1$) having elements

$$w_{ki} \triangleq \sum_{u=1}^Q \sum_{s=1}^{N-k} \alpha_{iu}^{(s)} \alpha_{iu}^{(s+k)} q_u; \quad i = 1, \dots, I$$

as the codeword contributions to ρ_k (the autocorrelation coefficient).

The element w_{ki} of \underline{W}_k is called the contribution to ρ_k by state i . These definitions allow us to express v'_k as

$$v'_k = p \underline{W}_k - \sum_{s=1}^{N-k} \mu(s) \mu(s+k). \quad (2.3)$$

The determination of v''_k is complicated by the fact that the state transition must be considered. Assuming the encoder to be in state i , we have the conditional expectation

$$\sum_{j=1}^I \sum_{u \in B_{ij}} \sum_{s=N-k+1}^N \sum_{v=1}^Q ([\alpha_{iu}^{(s)} - \mu^{(s)}][\alpha_{jv}^{(s+k-N)} - \mu^{(s+k-N)}] q_u q_v) .$$

After averaging over all states and rearranging terms we obtain

$$v_k'' = \sum_{s=N-k+1}^N \left[\left\{ \sum_{i=1}^I \sum_{j=1}^I p_i \left[\sum_{u \in B_{ij}} q_u \alpha_{iu}^{(s)} \right] \left[\sum_{v=1}^Q q_v \alpha_{jv}^{(s+k-N)} \right] \right\} - (\mu^{(s)} \mu^{(s+k-N)}) \right] .$$

We can substitute the matrix notation developed earlier, which yields

$$v_k'' = \sum_{s=N-k+1}^N \left[p_s^M M_{s+k-N} \underline{u} - (p_s^M \underline{u})(p_{s+k-N}^M \underline{u}) \right] . \quad (2.4)$$

However $\underline{u} \cdot \underline{p} = \Pi_\infty$, and since $\Pi^0 = U$ (the identity matrix) we finally arrive at

$$v_k'' = \sum_{s=N-k+1}^N p_s^M (\Pi^0 - \Pi_\infty) M_{s+k-N} \underline{u} . \quad (2.5)$$

Hence, upon combining (2.3) and (2.4), we have

$$v_k = \frac{1}{N} \left\{ p \underline{W}_k - \sum_{s=1}^{N-k} \mu^{(s)} \mu^{(s+k)} + \sum_{s=N-k+1}^N p_s^M (\Pi^0 - \Pi_\infty) M_{s+k-N} \underline{u} \right\}$$

for $k = 1, \dots, N-1$.

For the remaining covariance coefficients, v_k ; $k \geq N$, the same notation may be used. Consider Fig. 2 below representing the encoded sequence.

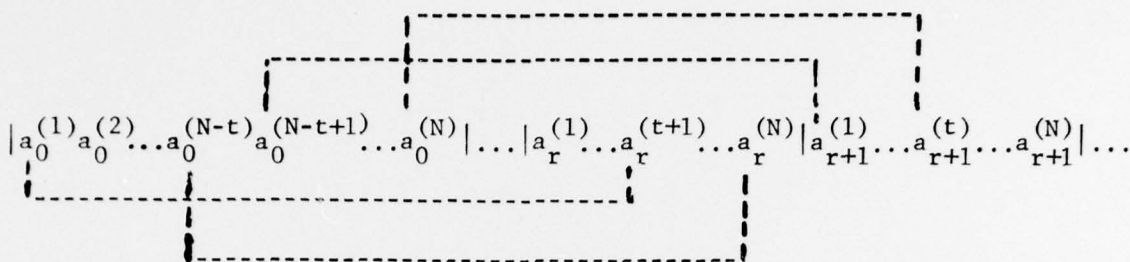


Figure 2.

The dotted lines indicate the extremes of symbols involved in the calculation of $v_k = v_{rN+t}$. In other words,

$$v_{rN+t} = \frac{1}{N} \left\{ \sum_{s=1}^{N-t} E \left([a_0^{(s)} - \mu^{(s)}] [a_r^{(s+t)} - \mu^{(s+t)}] \right) + \sum_{s=N-t+1}^N E \left([a_0^{(s)} - \mu^{(s)}] [a_{r+1}^{(s+t-N)} - \mu^{(s+t-N)}] \right) \right\}.$$

Both of the sums in this expression are of the same form and the expectations can be represented as in (2.5) with an additional sum which takes into account the number of state transitions occurring between a_0 and a_r for the first sum, and a_0 and a_{r+1} in the second sum. With this in mind, we can immediately obtain

$$v_{rN+t} = \frac{1}{N} \left\{ \sum_{s=1}^{N-t} p M_s (\Pi^{r-1} - \Pi_{\infty}) M_{s+t} \underline{u} + \sum_{s=N-t+1}^N p M_s (\Pi^r - \Pi_{\infty}) M_{s+t-N} \underline{u} \right\}.$$

It must be noted that when $t = 0$ (i.e., for covariance coefficients at integral multiples of N), the second term is understood to be zero.

The expressions for the covariance coefficients are summarized in Table 1 below.

Table 1. Covariance Coefficients

k	v_k
0	$\frac{1}{N} [p \underline{w}_0 - \sum_{s=1}^N (\mu^{(s)})^2]$
1, ..., N-1	$\frac{1}{N} \left(p \underline{w}_k - \sum_{s=1}^{N-k} \mu^{(s)} \mu^{(s+k)} + \sum_{s=N-k+1}^N p M_s (\Pi^0 - \Pi_\infty) M_{s+k-N} \underline{u} \right)$
$rN; r \geq 1$	$\frac{1}{N} \sum_{s=1}^N p M_s (\Pi^{r-1} - \Pi_\infty) M_s \underline{u}$
$rN+t;$ $r \geq 1,$ $t=1, \dots, N-1$	$\frac{1}{N} \left(\sum_{s=1}^{N-t} p M_s (\Pi^{r-1} - \Pi_\infty) M_{s+t} \underline{u} + \sum_{s=N-t+1}^N p M_s (\Pi^r - \Pi_\infty) M_{s+t-N} \underline{u} \right)$

III. Z-Transform Approach to Spectral Analysis

In its most general form Bennett's formula for power spectral density, $S_x(f)$, for the discrete encoded process is as follows:

$$S_x(f) = v_0 + 2 \sum_{k=1}^{\infty} v_k \cos 2\pi f k + \frac{1}{N} \left| \sum_{i=1}^N \mu^{(i)} e^{-j2\pi i f} \right|^2 \sum_{l=-\infty}^{\infty} \delta(f - \frac{l}{N}) \quad (3.1)$$

where v_k and $\mu^{(i)}$ are the covariance coefficients and digit mean values respectively. The $\delta(\cdot)$ is the Dirac impulse function.

The spectrum, seen to be composed of a continuous portion (denoted as $X_c(f)$) and a discrete portion (denoted $X_d(f)$), is periodic with unity period. Note that the discrete lines are spaced at multiples of $1/N$.

Since the covariance sequence is even, it is easily seen that the z-transform of the sequence $\{v_k\}$ can be expressed as $R(z) + R(z^{-1})$ where $R(z) \triangleq \frac{v_0}{2} + \sum_{k=1}^{\infty} v_k z^{-k}$. Furthermore, it should be apparent that

$$X_c(f) = 2\text{Re}\{R(e^{j2\pi f})\} \quad (3.2)$$

For the next step in the derivation, we shall express $R(z)$ in terms of the matrix expressions for the covariance coefficients developed in Section II.

It is convenient for us to express $R(z)$ as follows:

$$R(z) = \frac{v_0}{2} + \sum_{k=1}^{N-1} v_k z^{-k} + \sum_{r=1}^{\infty} v_{rN} z^{-rN} + \sum_{i=1}^{\infty} \sum_{j=1}^{N-1} v_{iN+j} z^{-(iN+j)} \quad (3.3)$$

Upon making the appropriate substitutions from Table 1 into (3.3), we obtain

$$\begin{aligned}
N \cdot R(z) = & \frac{1}{2} \left[\underline{pW}_0 - \sum_{s=1}^N [\mu(s)]^2 \right] + \sum_{k=1}^{N-1} \left\{ \underline{pW}_k - \sum_{s=1}^{N-k} \mu(s) \mu(s+k) \right. \\
& + \sum_{s=N-k+1}^N \underline{pM}_s (\Pi^0 - \Pi_\infty) M_{s+k-N} \underline{u} \left. \right\} z^{-k} \\
& + \sum_{r=1}^{\infty} \sum_{s=1}^N \left[\underline{pM}_s (\Pi^{r-1} - \Pi_\infty) M_s \underline{u} \right] z^{-rN} \\
& + \sum_{t=1}^{\infty} \sum_{k=1}^{N-1} \left[\sum_{s=1}^{N-k} \underline{pM}_s (\Pi^{t-1} - \Pi_\infty) M_{s+k} \underline{u} \right. \\
& \left. + \sum_{s=N-k+1}^N \underline{pM}_s (\Pi^t - \Pi_\infty) M_{s+k-N} \underline{u} \right] z^{-(tN+k)}.
\end{aligned}$$

After rearranging terms and changing the index of summation t , we arrive at

$$\begin{aligned}
N \cdot R(z) = & \frac{1}{2} \left[\underline{pW}_0 - \sum_{s=1}^N [\mu(s)]^2 \right] + \sum_{k=1}^{N-1} \left[\underline{pW}_k - \sum_{s=1}^{N-k} \mu(s) \mu(s+k) \right] z^{-k} \\
& + z^{-N} \sum_{r=0}^{\infty} \sum_{k=0}^{N-1} \left[\sum_{s=1}^{N-k} \underline{pM}_s (\Pi^r - \Pi_\infty) M_{s+k} \underline{u} \right] z^{-(rN+k)} \\
& + \sum_{i=0}^{\infty} \sum_{k=1}^{N-1} \left[\sum_{s=N-k+1}^N \underline{pM}_s (\Pi^i - \Pi_\infty) M_{s+k-N} \underline{u} \right] z^{-(iN+k)}. \quad (3.4)
\end{aligned}$$

In order to simplify the notation we define N constants as follows:

$$\omega_k = \begin{cases} \frac{1}{2} \{ \underline{pW}_0 - \sum_{s=1}^N [\mu^{(s)}]^2 \}; & k=0 \\ \underline{pW}_k - \sum_{s=1}^{N-k} \mu^{(s)} \mu^{(s+k)}, & k=1, \dots, N-1. \end{cases} \quad (3.5)$$

Further simplification of (3.4) is possible since only the state transition matrix, Π , and z are dependent upon the summation indices i and r . After the appropriate rearrangement we obtain

$$\begin{aligned} N \cdot R(z) &= \sum_{k=0}^{N-1} \omega_k z^{-k} + z^{-N} \sum_{k=0}^{N-1} \sum_{s=1}^{N-k} \underline{pM}_s \left(\sum_{r=0}^{\infty} [\Pi^r - \Pi_{\infty}] z^{-rN} \right) M_{s+k} \underline{u} z^{-k} \\ &+ \sum_{k=1}^{N-1} \sum_{s=N-k+1}^N \underline{pM}_s \left(\sum_{i=0}^{\infty} [\Pi^i - \Pi_{\infty}] z^{-iN} \right) M_{s+k-N} \underline{u} z^{-k}. \end{aligned} \quad (3.6)$$

Following the development given in [5], we have the equalities

$$\sum_{r=0}^{\infty} [\Pi^r - \Pi_{\infty}] z^{-rN} = z^N (U - \Pi_{\infty}) [z^N U - (\Pi - \Pi_{\infty})]^{-1} = (U - \Pi_{\infty}) G/D(z^N)$$

where $\lambda D(\lambda) = \lambda \sum_{k=0}^{I-1} d_{I-k-1} \lambda^k$, ($d_0 = 1$), is the characteristic polynomial

of $\Pi - \Pi_{\infty}$ and G is called the conjoint matrix of $\Pi - \Pi_{\infty}$. The conjoint matrix can be written as a polynomial with matrix coefficients of the form

$$G = \sum_{k=0}^I G_{I-k} z^{N(k-1)}, \quad (G_I = 0, G_0 = U). \quad (3.7)$$

Substitution of (3.7) into (3.6) and some algebraic manipulation leads to

$$\begin{aligned} N \cdot R(z) = & \sum_{k=0}^{N-1} \omega_k z^{-k} + p \left[\sum_{t=1}^I \left[\sum_{k=0}^{N-1} \sum_{s=1}^{N-k} M_s \hat{G}_{I-t} M_{s+k} z^{Nt-k-1} \right. \right. \\ & \left. \left. + \sum_{k=1}^{N-1} \sum_{s=N-k+1}^N M_s \hat{G}_{I-t} M_{s+k-N} z^{N(t+1) - k-1} \right] \underline{u} / z^{2N-1} D(z^N) \right] \end{aligned} \quad (3.8)$$

where $\hat{G}_{I-t} \triangleq (U - \Pi_\infty) G_{I-t}$.

The polynomial in the numerator of the second term is of degree $(I+1)N-2$. In order to develop a convenient method of computation for the scalar coefficients, we shall define two sets of vectors as follows:

$$\begin{aligned} 1) \quad \underline{\eta}_{ts} & \triangleq p M_s \hat{G}_{I-t}; \quad t = 1, \dots, I \text{ and } s = 1, \dots, N, \\ 2) \quad \underline{\xi}_s & = M_s \underline{u}; \quad s = 1, \dots, N. \end{aligned} \quad (3.9)$$

The $NI \underline{\eta}_{ts}$ are seen to be $(1 \times I)$ row vectors and the $\underline{\xi}_s$ are $(I \times 1)$ column vectors. Substituting these vectors into (3.8) yields

$$\begin{aligned} N \cdot R(z) = & \sum_{k=0}^{N-1} \omega_k z^{-k} + \left[\sum_{t=1}^I \left(\sum_{k=0}^{N-1} \sum_{s=1}^{N-k} \underline{\eta}_{ts} \underline{\xi}_{s+k} z^{Nt-k-1} \right. \right. \\ & \left. \left. + \sum_{k=1}^{N-1} \sum_{s=N-k+1}^N \underline{\eta}_{ts} \underline{\xi}_{s+k-N} z^{N(t+1) - k-1} \right) \right] / z^{2N-1} D(z^N). \end{aligned} \quad (3.10)$$

For a fixed t , consider the convolution of the two vector sequences

$(\underline{\eta}_{t1}, \dots, \underline{\eta}_{tN})$ and $(\underline{\xi}_1, \dots, \underline{\xi}_N)$, which we shall call the sequence

$\underline{c}_t = (c_{t1}, \dots, c_{t, (2N-1)})$. Furthermore, consider the coefficients of the

numerator polynomial in (3.10). For a fixed t we can express the coefficient of z^{Nt-k-1} for $k \neq 0$ as,

$$\sum_{s=N-k+1}^N \eta_{ts} \xi_{s+k} + \sum_{s=1}^{N-k} \eta_{(t-1)s} \xi_{s+k-N}.$$

However the first sum equals $C_{t(N-k)}$ and the second sum equals $C_{(t-1)(N+k-1)}$.

When $k = 0$ the coefficient of z^{Nt-1} is simply C_{tN} .

IV. ALGORITHM

The development of the expression for the power spectral density of the random process generated by a finite-state machine encoder allows us to design an algorithm for the efficient computation of both the discrete and continuous spectrum. We shall describe the algorithm in three segments: input processing, main program, and the point-by-point evaluation. The input processing consists of reading the codebook, state transition table and the probability distribution for the binary words. This data is used to calculate the state transition matrix, Π , the N transition weight matrices, M_s , and the N vectors of codeword contributions \underline{W}_k . The main program uses these quantities to calculate the coefficients of the polynomials which determine the discrete and continuous spectrum. Finally, the point-by-point evaluation calculates the continuous spectrum at any desired frequency, and evaluates the weights of the spectral lines.

The codebook is input one codeword at a time with the following format:

- (a) codeword \underline{a} (N integers),
- (b) codeword's state, $i \in \mathcal{S}$,
- (c) the state transition, $j \in \mathcal{S}$,
- (d) the probability, q_u , of binary input $b_u \in \mathcal{S}$
such that $f_i(b_u) = \underline{a}$.

For each word, the following calculations must be made:

- (a) $\pi_{ij} \leftarrow \pi_{ij} + q_u$
- (b) $m_{ij}^{(s)} \leftarrow m_{ij}^{(s)} + q_u \alpha_{iu}^{(s)}$; $s = 1, \dots, N$
- (c) $w_{ki} \leftarrow w_{ki} + q_u \sum_{s=1}^{N-k} \alpha_{iu}^{(s)} \alpha_{iu}^{(s+k)}$; $k = 0, \dots, N-1$.

Note that in general, $I \cdot 2^K$ codewords will be processed in this manner.

The calculations performed in the main program segment may be summarized as follows:

- (a) computation of \underline{p} from $\underline{p}\Pi = \underline{p}$ and $\sum_{i=1}^I p_i = 1$, and construction of Π_{∞} ,
- (b) computation of the vectors $\underline{\xi}_s = M_s \underline{u}$; $s = 1, \dots, N$,
- (c) computation of $\mu^{(\ell)} = \underline{p} M_{\ell} \underline{u} = \underline{p} \underline{\xi}_{\ell}$; $\ell = 1, \dots, N$,
- (d) computation of w_k ; $k = 0, \dots, N-1$ from (3.5),
- (e) computation of the matrix coefficients, G_{I-k} , $k = 0, \dots, I-1$, and the scalar coefficients d_{I-k} ; $k = 0, \dots, I$. This can be done recursively as follows (see [5]):

(i) let $F \triangleq \Pi - \Pi_{\infty}$, then

$$d_k = -\frac{1}{k} \text{trace} \{FG_{k-1}\}, \text{ and,}$$

$$G_k = FG_{k-1} + d_k U,$$

where $k = 1, \dots, I$, with initial conditions

$$d_0 = 1 \text{ and } G_0 = U \text{ (the identity). A check}$$

can be made on these calculations since

$$d_I = 0 \text{ and } G_I = \|0\|,$$

- (f) computation of the vectors $\underline{\eta}_{ts} \triangleq \underline{p} M_s \hat{G}_{I-t} = \underline{p} M_s (U - \Pi_{\infty}) G_{I-t}$,

for $t = 1, \dots, I$, and $s = 1, \dots, N$ (note that one matrix multiplication can be saved since $G_0 = U$),

- (g) computation of the constants C_{tj} , $t = 1, \dots, I$ and

$j = 1, \dots, 2N-1$ by performing the convolutions

$$\eta_t * \xi \text{ for } t = 1, \dots, I, \text{ where } \eta_t \text{ and } \xi \text{ are the vector sequences,}$$

- (h) computation of the coefficients a_j , $j = 0, \dots, (I+1)N-2$.

This may be done by considering the I sequences, \underline{c}_t

as polynomials of the form

$$C_t(z) = \sum_{i=0}^{2N-2} C_{t,i+1} z^i.$$

The numerator polynomial may then be expressed as

$$\sum_{t=1}^I z^{N(t-1)} C_t(z).$$

This implies that the sequence of numerator coefficients may be found by right-shifting C_t by $N(t-1)$ units and summing the set of shifted sequences.

The point-by-point evaluation segment of the algorithm computes the continuous spectral density for any frequency, f , from

$$X_c(f) = 2\text{Re} \left\{ \left(\sum_{k=0}^{N-1} \omega_k z^{-k} + \frac{\sum_{i=0}^{(I+1)N-2} a_i z^i}{z^{2N-1} D(z^N)} \right) \right\} \Big|_{z = e^{j2\pi f}}.$$

This is best accomplished by evaluating

$$X_1(f) \triangleq \sum_{k=0}^{N-1} \omega_k z^{-k}, \quad X_2(f) \triangleq \frac{\sum_{i=0}^{(I+1)N-2} a_i z^i}{z^{2N-1} D(z^N)},$$

and $D(z^N)$ for $z = e^{j2\pi f}$, and then combining the result as

$$X_c(f) = 2\text{Re}\{X_1(f) + X_2(f)/z^{2N-1} D(z^N)\}.$$

The discrete spectrum is found from

$$X_d(f) = \frac{1}{N} \left| \sum_{i=1}^N \mu^{(i)} z^{-i} \right|^2; \quad z = e^{j \frac{2\pi \ell}{N}}, \quad \ell = 0, \dots, N-1.$$

V. PERFORMANCE EVALUATION (COMPLEXITY ANALYSIS)

In order to determine the computational efficiency of this algorithm, we shall analyze the amount of data storage necessary, and the number of scalar multiplications needed to complete the computation of the power spectral density.

The data storage (not including work space) is itemized in Table 2.

Table 2

<u>Variable</u>	<u>Storage Space (Integers)</u>
$M_s; s=1, \dots, N$	$I^2 N$
Π, Π_∞	$2I^2$
$G_\ell; \ell=1, \dots, I$	I^3
$\mu^{(\ell)}; \ell=1, \dots, N$	N
P	I
$\underline{W}_k; k=0, \dots, N-1$	NI
$d_k; k=1, \dots, I$	I
$\eta_{ts}; t=1, \dots, I$ $s=1, \dots, N$	$I^2 N$
$\underline{E}_s; s=1, \dots, N$	IN
$a_j; j=0, \dots, (I+1)N-2$	$IN + N-1$

Adding the memory requirements shown in Table 2 yields a total of

$$I^3 + I^2(2N+2) + I(3N+2) + 2N-1. \quad (5.1)$$

The number of scalar multiplications will be counted for each segment of the algorithm. The input processing segment requires multiplications for

steps (b) and (c). Step (b) requires N scalar multiplications for each codeword. For step (c), the sums

$$q_u = \sum_{s=1}^{N-k} \alpha_{iu}^{(s)} \alpha_{iu}^{(s+k)}$$

must be calculated for $k=0, \dots, N-1$. This takes

$$\frac{N(N+1)}{2} + N = \frac{N^2 + 3N}{2}$$

multiplications for each codeword. Hence, the total number of multiplications for the input processing segment is

$$2^{K-1} \cdot I(N^2 + 5N + 1). \quad (5.2)$$

The multiplications necessary to complete the main program are summarized as follows (with reference to Section IV):

- (a) $\gamma(I+1)^3$; for a constant γ depending on the matrix inversion algorithm used,
- (b) none, because no multiplications are necessary for the matrix-vector product $M_s \underline{u}$,
- (c) NI , (N inner products with vectors of dimension I),
- (d) the N inner products of the form $\underline{p} \cdot \underline{W}_k$ require NI multiplications, while the sums

$$\sum_{s=1}^{N-k} \mu_{\mu}^{(s)} \mu_{\mu}^{(s+k)}; \quad k=0, \dots, N-1$$

require $\frac{1}{2}(N^2 + N)$ multiplications, yielding a total of

$$NI + \frac{1}{2}(N^2 + N),$$

(e) each coefficient d_k , $k=1, \dots, I$ requires $I^3 + 1$ and the matrix coefficients do not require any more, giving a total of $I^4 + I$ multiplications,

(f) the vectors \underline{u}_{ts} require N products of the form $\underline{p}_s^M (U - \underline{u}_\infty) \triangleq \underline{x}$, and for each such vector, there are I products of the form $\underline{x} G_{I-t}$. This yields a total of $2I^2N + I^3N$ multiplications,

(g) each vector convolution requires N^2 inner products of vectors having dimension I , and since there are I convolutions, we need to perform N^2I^2 scalar multiplications for this step.

Since step (h) doesn't require any multiplications, we obtain a total of

$$I^4 + \gamma(I+1)^3 + I^3N + I^2(N^2+2N) + I(2N+1) + \frac{1}{2}(N^2+N). \quad (5.3)$$

The point-by-point evaluation for the continuous spectrum requires the evaluation of 3 polynomials (complex); two of degree $(N-1)$ and one of degree $(I+1)N-2$. Using Horner's Rule, these evaluations require a total of $IN + 3N - 4$ complex multiplications (or $4IN + 12N - 16$ scalar multiplications) for each value of frequency.

The discrete spectrum also requires the evaluation of an $(N-1)^{\text{th}}$ degree polynomial at $\lfloor N/2 \rfloor$ points. In addition, each point requires 2 multiplications to obtain the magnitude-squared value. Thus the total number of scalar multiplications needed for the discrete spectrum is

$$\lfloor N/2 \rfloor (4N - 2).$$

Since the total complexity of the spectrum analysis program is sensitive to the number of frequencies (points) desired, we shall only consider the input processing and main program segments. Thus, upon adding (5.2) and (5.3) we arrive at

$$2^{K-1} \cdot I(N^2+5N) + I^4 + \gamma(I+1)^3 + I^3N + I^2(N^2+2N) + I(2N+1) + \frac{1}{2}(N^2+N) \quad (5.4)$$

for the total number of scalar multiplications necessary to execute the algorithm.

A similar analysis of the algorithm given by Cariolaro and Tronca in [5] yields

$$2^K [IN+I^2+1] + I^3 + 2I^2 + I(N^2+3N+2) + 2N^2 + N - 2 \quad (5.5)$$

as the data storage required, and

$$2^K [3I^2N+I^2+I(N^2+N)] + I^4 + \gamma(I+1)^3 + 2NI^3 + I^2(N^2+N) + I(N^2+1) \quad (5.6)$$

as the total number of scalar multiplications (ignoring the point-by-point evaluation which is the same for both algorithms).

Comparison of (5.1) with (5.5) and (5.4) with (5.6) reveals that the algorithm based on digit sums presented here is more efficient than [5] for all (K, N, I) codes in both the amount of data storage and number of multiplications. An intuitive explanation of this increased efficiency is based on the amount of data used in the algorithm described in [5]; all of the codewords are used in all phases of the computation. However, it is seen from the analysis in Section II that all of this information is not necessary to determine the covariance of the encoded process, hence a more efficient algorithm.

References

- 1) Bennett, W. R., "Statistics of Regenerative Digital Transmission," B.S.T.J., Vol. 37, No. 6, November, 1958, pp. 1501-1542.
- 2) Tsujii, S., and Kasai, H., "An Algebraic Method of Computing the Power Spectrum of Coded Pulse Trains," Elect. & Comm. in Japan, Vol. 51-A, No. 7, July, 1968, pp. 36-44.
- 3) Yasuda, H., "Calculation of Power Spectra of Pulse Sequences by Means of Transition-Probability Matrices," Elect & Comm. in Japan, Vol. 54-A, No. 11, November, 1971, pp. 1-7.
- 4) Bosik, B. S., "The Spectral Density of a Coded Digital Signal," B.S.T.J., Vol. 51, No. 4, April, 1972, pp. 921-932.
- 5) Cariolaro, G. L., and Tronca, G. P., "Spectra of Block Coded Digital Signals," IEEE Trans. on Comm., Vol. COM-22, No. 10, October, 1974, pp. 1555-1563.
- 6) Preparata, F. P., and Bellato, L., "Length-2 Alphabetic Codes for Spectrum Shaping," Proc. from 1974 Zurich Seminar on Digital Communications, pp. F2(1)-F2(6).